```
  args1(0).Name = "Flags"
  args1(0).Value = "A"
  args1(1).Name = "FormulaCommand"
  args1(1).Value = 3
  args1(2).Name = "SkipEmptyCells"
  args1(2).Value = false
  args1(3).Name = "Transpose"
  args1(3).Value = false
  args1(4).Name = "AsLink"
  args1(4).Value = false
  args1(5).Name = "MoveMode"
  args1(5).Value = 4

  dispatcher.executeDispatch(document, ".uno:InsertContents", "", 0, 
args1())

end sub
```

**Note**

The component model used in LibreOffice is Universal Network Objects (UNO) and the macro recorder uses the UNO dispatcher for most commands. However, there are two problems associated with this technical approach. One is that the dispatches are not fully documented and may be subject to change. Another is that the recorder ignores some values from dialogs that are opened while recording a macro – it is possible, therefore, that you will record a complicated macro that will not actually execute everything as expected. For more information, search for “macro recording – limitations” in the Help index.

# Write your own functions

## Create function macro

You can write a macro and then call it as you would call a Calc function. Use the following steps to create a simple function macro:

1) Use **File > New > Spreadsheet** on the Menu bar to create a new spreadsheet, save it with the name CalcTestMacros.ods, and leave it open in Calc.
2) Use **Tools > Macros > Organize Macros > Basic** on the Menu bar to open the Basic Macros dialog (Figure 432). Note that the layout of the Basic Macros dialog in this circumstance is different from the version that Calc displays when the user clicks the **Stop Recording** button on the Record Macro dialog (see Figure 429).